Multicast routing with bandwidth requirement in the case of incomplete information as a Steiner tree problem

Patrice Leguesdron, János Levendovszky, Miklos Molnar, Csaba Végsö

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Multicast routing with bandwidth requirement in the case of incomplete information as a Steiner tree problem

Patrice Leguesdron, János Levendovszky, Miklós Molnár, Csaba Végsö

N°4343

Décembre 2001

THÈME 1

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--4343--FR+ENG

Multicast routing with bandwidth requirement in the case of incomplete information as a Steiner tree problem

Patrice Leguesdron*, János Levendovszky†, Miklós Molnár‡, Csaba Végsö§

Thème 1 — Réseaux et systèmes
Projet ARMOR

Rapport de recherche n°4343 — Décembre 2001 — 18 pages

Abstract: This paper is concerned with developing novel algorithms for multicast routing in packet switched communication networks. First, multicast routing with bandwidth requirement in the case of incomplete information is reduced to a deterministic Steiner tree problem. Then taboo search algorithms are used to provide high quality, sub-optimal solutions for multicast routing in polynomial time.

Key-words: Steiner tree problem, multicast networking, taboo search, enumeration algorithm

(Résumé : tsvp)

* INSA-IRMAR, Patrice.Leguesdron@insa-rennes.fr
† TU Budapest, levendov@hit.bme.hu
‡ INSA - IRISA, Miklos.Molnar@irisa.fr
§ TU Budapest, VEGSOCS@hit.hit.bme.hu

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : 02 99 84 71 00 - International : +33 2 99 84 71 00
Télécopie : 02 99 84 71 71 - International : +33 2 99 84 71 71

Routage multicast avec information incomplète sur la bande passante vu comme un problème de Steiner

Résumé : Dans cet article, nous développons un nouvel algorithme pour le routage multicast dans les réseaux de communication. Nous transformons le problème du routage multicast avec contrainte de bande passante lorsque l'information est incomplète en un problème de Steiner déterministe. Les algorithmes de la méthode "tabou" sont ensuite utilisés pour s'assurer de la bonne qualité des solutions sous optimale du routage multicast en un temps polynomial.

Mots-clé : Problème de Steiner, routing multicast, recherche "tabou", algorithme d'énumération

1 Introduction

Due to the growing need for multicast applications (e.g., video conference, video-on-demand, multiplayer games ... etc), Multicast Routing (MR) tends to play an instrumental role in modern networking. In order to provide real-time applications, one has to find the optimal spanning tree of multicast nodes, which fulfill given Quality of Service (QoS) criteria. There are several criteria which are associated with QoS. One of the most important ones is bandwidth. In this case, one must ensure that each link of the spanning tree need to have the minimal bandwidth required by the specific application. Furthermore, bandwidth of each link can vary according to the random traffic load presented in the network. Unicast Routing (UR) with uncertain link parameters was discussed in [5, 9]. As a result, MR can be viewed as random graph optimization problem, which needs novel techniques being different from traditional methods.
In this paper, first we demonstrate that MR can be reduced to a deterministic Steiner tree problem. Then we develop a suboptimal, but polynomial solution to obtain the optimal Steiner tree by using taboo search algorithms. The results are tested by extensive simulations. These topics will be treated in the following structure:

- in *Section 2* the model and the notation is introduced to tackle the problem;
- in *Section 3* we demonstrate how multicast routing, in the case of incomplete information, can be reduced to Steiner tree problem if the link measure is bandwidth;
- in *Section 4* some traditional enumeration algorithms are detailed to find the optimal Steiner tree;
- in *Section 5* the technique of taboo search is used to solve the Steiner tree problem;
- in *Section 6* an extensive numerical analysis of the methods are given.

2 The model

Let the network be modeled by a non-oriented, connected graph $G : (V, E, \delta)$, where V, E denote the sets of nodes and edges, respectively. The link measures $\delta_{(u,v)}$, $(u,v) \in E$ (e.g. bandwidths) are random variables subject to the probability distribution $F_{(u,v)}(x) = P\left(\delta_{(u,v)} < x\right)$. Let $R \subset V$ be a subset of multicast nodes to be spanned by the partial tree. Traditionally, any partial spanning tree would suffice which fulfils the criterion:

$$T_{solution} : \min_{(u,v)\in T} \delta_{(u,v)} > W, \tag{1}$$

where W is the the bandwidth requirement. However, due to the fact that $\delta_{(u,v)}$, $(u,v) \in E$ are random variables, expression (1) can only satisfy with a certain probability. Therefore the optimal tree must be selected based on the criterion given bellow:

$$T_{opt} : \max_{T\in\mathcal{T}} P\left(\min_{(u,v)\in T} \delta_{(u,v)} > W\right). \tag{2}$$

We term this problem as Maximum Likelihood Tree Selection (MLTS) and our goal is to develop solutions which can be given in polynomial time.

3 MLTS as a Steiner problem

In this section we reduce MLTS to a classical Steiner network problem, the solution of which can then be given by heuristics published in [7] and [13]. In order to achieve this goal we prove the following theorem:

Theorem. *MLTS reduces to a Steiner tree problem by using the link metric*

$$\kappa_{(u,v)} := -\log P(\delta_{(u,v)} > W).$$

Proof. Evaluating the probability

$$P\left(\min_{(u,v)\in T} \delta_{(u,v)} > W\right),$$

one can rewrite it into the form

$$P\left(\min_{(u,v)\in T} \left(\delta_{(u,v)} > W\right)\right) = P\left(\bigcap_{(u,v)\in T} \delta_{(u,v)} > W\right).$$

Assuming link independence, this can be reformulated as

$$\begin{aligned} \textstyle\prod_{(u,v)\in T} P\left(\delta_{(u,v)} > W\right) &= e^{\log\left(\prod_{(u,v)\in T} P\left(\delta_{(u,v)}>W\right)\right)} \\ &= e^{\log\left(\prod_{(u,v)\in T} P\left(\delta_{(u,v)}>W\right)\right)} \\ &= e^{\sum_{(u,v)\in T} \log\left(P\left(\delta_{(u,v)}>W\right)\right)}. \end{aligned}$$

As the exponential function is monotone, the original optimization task

$$T_{opt} : \max_{T\in\mathcal{T}} P\left(\min_{(u,v)\in T} \delta_{(u,v)} > W\right)$$

reduces to

$$T_{opt} : \max_{T\in\mathcal{T}} \sum_{(u,v)\in T} \log\left(P\left(\delta_{(u,v)} > W\right)\right).$$

Furthermore, since $-\log P\left(\delta_{(u,v)} > W\right) \geq 0$, finding the solution to MLTS reduces to

$$T_{opt} : \max_{T\in\mathcal{T}} \sum_{(u,v)\in T} \log\left(P\left(\delta_{(u,v)} > W\right)\right), \tag{3}$$

which is indeed a Steiner tree problem with link metric $\kappa_{(u,v)} := -\log P\left(\delta_{(u,v)} > W\right)$.
Q.E.D.

4 Two particular Steiner tree enumeration algorithms

Reducing MLTS to a Steiner tree problem, we now embark on proposing two enumeration algorithms, which can yield a suboptimal solution in polynomial time. In sub-sections 4.2 and 4.3, the principles of two search algorithms for the optimal solution of the Steiner problem are presented. These principles will be used to formulate our proposal for taboo search. But before searching for exact or approximate solutions, some reductions proposed by Balakrishnan and Patel of the graph can be done. In the following sub-section we discuss these reductions.

4.1 Graph reductions

To reduce the complexity of the Steiner problem in the case of an arbitrary graph $G = (V, E)$, Balakrishnan and Patel proposed various reductions ([1], [10]). Reductions can precede the search procedure for partial spanning tree with minimal length. Balakrishnan and Patel classify nodes and edges in the following way:

- a node is known as R-type node if it is included into the set R of nodes to be covered,
- a node is known as S-type node if it belongs to the set $S = V \backslash R$,
- the edge $(i, j) \in E$ is known as
  - . R-R edge, if $i \in R$ and $j \in R$,
  - . S-S edge, if $i \in S$ and $j \in S$,
  - . R-S edge, if one of its ends belongs to R and the other to S.

Initial reductions are based on minimal Steiner tree properties:

i) S-type leaves of G have no consequence on the optimal solution and can be eliminated.

ii) A path with R-type end points containing only S-type internal nodes of degree two can be replaced by one R-R edge. This edge must have same length than the substitued path.

iii) In the search for spanning tree, R-type leaves are necessarily connected to their neighbour. Consequently, if the graph G contains R-type leaves, these leaves can be united with their neighbour node and this new node becomes of R-type.

iv) If, for two adjacent nodes, there exists a shorter path than the edge connecting them, then this edge can be eliminated from graph G.

The second aspect analysed by Balakrishnan and Patel is related to the connections between various spanning trees associated with different sub-graphs of G. Four types of reductions are proposed.

Proposal BP_I: aggregation of nodes. Let $S^* \subset S$ be the subset of S-type nodes in the solution. The minimal Steiner tree must contain all the edges of $MST(G)$ connecting two nodes of $R \cup S^*$.

Aggregation of R-type nodes. According to the BP_I proposal, each R-R edge of $MST(G)$ belongs also to the minimal Steiner tree. Consequently, two R-type adjacent nodes in $MST(G)$ can be grouped together in one R-type node to obtain an equivalent problem.

Balakrishnan and Patel also state that agregation of nodes can introduce parallel edges. In this case, to find an equivalent 1-graph, only the shortest edge among the parallel edges must be kept.

Restriction of R-S type edges. Let us suppose that $MST(G)$ contains an R-S type edge (i,j). Let us say $i \in S$ and $j \in R$. According to the BP_I proposal, the minimal Steiner tree contains (i,j), if and only if $i \in S^*$.

Proposal BP_{II}: removal of R-R type edges. Let $MST(R)$ be the minimal tree covering the graph generated by the set R. The minimal Steiner tree of the original problem does not contain an R-R type edge ouside $MST(R)$. If the graph generated by R is a connected graph, only $|R| - 1 = (r-1)$ edges of this graph (included $MST(R)$) must be kept to continue. Balakrishnan and Patel indicate that in a complete graph, this removal reduces the number of R-R edges by a factor $r/2$.

Proposal BP_{III}: removal of R-S type edges. An R-S type edge (i,j) of G with $i \in S$ and $j \in R$ can belong to the minimal Steiner tree of R over G, if and only if the minimal spanning tree $MST(R \cup \{i\})$ of the graph generated by the set $(R \cup \{i\})$ contains this edge.

Proposal BP_{IV}: removal of S-S type edges. S-S type edge (i,j) of G can belong to the minimal Steiner tree of R over G, if and only if the minimal spanning tree $MST(R \cup \{i\} \cup \{j\})$ contains this edge.

4.2 Balakrishnan and Patel algorithm for minimal Steiner tree construction

The first step of Balakrishnan and Patel's algorithm ([1]) introduces an artificial node a in the graph G. This node is connected to all the branching nodes of $V \backslash R$ and to an arbitrary node b of R. These artificial edges are null-length edges. A new graph $\widetilde{G}$ is thus obtained to search a spanning tree with minimal length which satisfy the following criterion (Balakrishnan and Patel): if the edge (a,b) is removed from the spanning tree, it remains a sub-tree covering all the nodes of R. Balakrishnan and Patel showed that in such a spanning tree of $\widetilde{G}$, the ends of the artificial edges different from a are leaves, if this artificial edge belongs to the spanning tree.

Figure 1 shows a graph G and the modified graph $\widetilde{G}$.

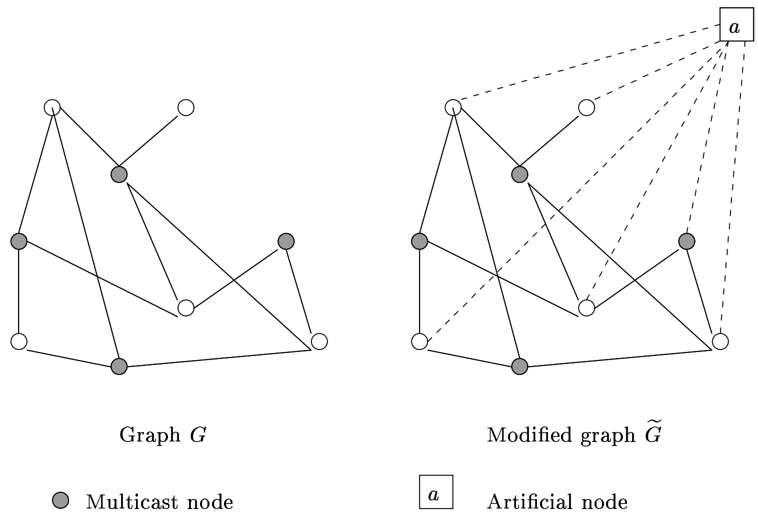

Figure 1:

To find the first solution, which satisfies this criterion, Balakrishnan and Patel use Gabow's algorithm ([4]). It provides the succession of spanning trees of the graph with in no decreasing length order. This solution is the partial minimal spanning tree of the arising problem. Balakrishnan and Patel's algorithm can be summarised as follows:

Initialisations
$\widetilde{G}$:= $G \cup \{a\}$ `and construct artificial edges as indicated`
$\widetilde{T}_1$:= $MST(\widetilde{G})$;
Iterations:
While `BP's condition is false` **do**
$\widetilde{T}_1$`:= the next tree created by Gabow's algorithm ;`
EndDo;
`Return the last tree (solution);`

4.3 A simple enumeration algorithm of Steiner trees

Molnar and Marie proposed a modification of Balakrishnan and Patel's algorithm [10]. In the modified algorithm, after completing the graph by the artificial part as suggested in Balakrishnan and Patel's method, all possible configurations composed with artificial edges are enumerated.

Let us define *configuration* as the set of artificial edges different from edge (a, b) which belong to the spanning tree.

A configuration will be called *acceptable* if the adjacent edges of the edges belonging to the configuration do not form a cut of G.

In Molnar and Marie's enumeration algorithm, all *acceptable* configurations of artificial edges are generated. The optimal solution is the tree with minimal length among the solutions obtained for *acceptable* configurations. Thus, the algorithm corresponds to an enumeration type algorithm, but the search is carried out on a restricted space only. This algorithm is described as follows:

Begin
    $Lgmin = +\infty$
// *Construction of the reduced graph* $G^{''}$ *from* $G^{'}$
    `Computation of MST(`$G^{'}$`)`
    `Let` $R^{``} \subset R$ `obtained after` BP_I `removal`
    If $R^{''}$ `connected` **then**
        `Calculate MST(`$R^{''}$`)`
        `Make all` BP_{II}`-removals`
        `Add an artificial node` a `connecting all` S`-type nodes`
        `(edge length is null)`
    EndIf
    `Choose a node` b
    O`:= sub-graph generated by artificial edges`
//
    For all `sub-tree` $O_i \subseteq O$ **do**
        `Determine a new graph` $G^{''}_{O_i}$ `obtained removing:`
            $i)$ `edges (null length) of` G'' `which are adjacent of`
                `leaves of` O_i `excepted the leaf` b
            $ii)$ `null length edges not in` O_i
        If $G^{''}_{O_i}$ `is connected,` **then**
            T_{O_i}`:= minimal spanning tree of` $G^{''}_{O_i}$
            If $d(T_{O_i}) < Lgmin$`,` **then**
                $T^*_O := T_{O_i}$
                $Lgmin := d(T_{O_i})$
            EndIf
            `Eliminate all excluded edges (proposals` BP_{III} `et` BP_{IV}`)`
        EndIf
    EndDo
End

Execution time of this exhaustive method for the exact solution can further be limited (and an approximate tree of the optimal solution can be obtained) by using the taboo search algorithm. This modification will be developed in the section 5.2.

5 Taboo search and Steiner trees

In this section we briefly revise the method of taboo search and then apply it to the Steiner tree problem. This method was applied succesfully to solve different network problems. For exemple: Taboo search has been used in [14, 15] for optimizing the link capacities in a dynamic telecommunication network.

5.1 Discrete optimization by Taboo search

Taboo search introduces the concept of memory in the course of searching the set of possible solutions. For each iteration, all possible elementary state transitions are examined and the "least wrong" is retained. Without memorising the successive state transitions, the algorithm may yield cycles. To avoid this problem, the concept of memory is used in order to bypass states which have already been visited. These prohibited solutions are referred to as "taboos". Currently the list of taboos is limited. The search of the "least wrong" solution can then be selected from the set from which the taboos are excluded.

To create a new solution X_{i+1} starting from the examined solution X_i, an *elementary transition* m is applied: $X_i \to X_{i+1}^m$.

The neighbourhood S (the set of the possible successors) of a solution X_i is defined as the set of solutions which can be generated from X_i using all possible elementary transition methods: $(X_i) = \{X_{i+1}^m, \forall m\}$.

The taboo method, which will be used in our treatment, is presented in the following way:

// Initialisations
  `Computation of a first solution` X
  X_{opt} `:=` X
  f_{min} `:=` $f(X)$
  $TabooList$ `:=` $\emptyset$
// Iterations
  While `stop condition is not true` **do**
    $C := S(X) \backslash \{$`transformations` $m(X)$ `from` X `with` $m^{-1} \in TabooList\}$
    `Choose` $Y = m(X)$ `minimising` f `in` C
    If $f(Y) \geq f(X)$ **then** `Add` m `to` $TabooList$
      `(Delete the oldest transformation eventually;)`
    EndIf
    If $f(Y) < f_{min}$ **then**
      $X_{opt} := Y$
      $f_{min} := f(Y)$
    EndIf
    $X := Y$
  Done
  `Return` X_{opt}

The stopping criterion can be the maximal number of iterations specified by the user.

5.2 Application of "taboo" into the Steiner tree enumeration algorithm

We take up the idea introduced by Balakrishnan and Patel: add to the graph G an artificial node a and a set of artificial edges. Here, the artificial edges which connect a to the nodes of $V \backslash R$ and to an arbitrary node of R have null or infinite length. To enumerate spanning trees, we will use the algorithm proposed by Molnar and Marie. To apply the "taboo" search method, we define a representation of the possible configurations of $\widetilde{G}$ and the elementary transformations which make possible transition from the current configuration X_i to a configuration X_j in the neighbourhood of X_i.

Representation of a configuration Let $|V \backslash R|$ be the number of nodes which do not belong to the group R (there are $|V \backslash R| + |R| + 1$ nodes in the graph $\widetilde{G}$). The number of artificial edges added to the graph G is decomposed into $|v \backslash R|$ edges connecting node a to the nodes of the set $V \backslash R$ and an edge connecting a to an arbitrary node of R. When enumerating the configurations, this last edge does not have any influence ; it always belongs to the spanning tree of $\widetilde{G}$ and its length will be always null.

We represent a configuration of the artificial edges as a binary coding set of $|V \backslash R|$ bits where each bit corresponds to an added edge. An edge will be coded by 1 (resp. 0) if it is (resp. not) taken into account in $\widetilde{G}$ for the search of the spanning tree. We can consider the length of the no influence edges in the search of the spanning tree is infinite.

Note: Since the graph G is connected, there is always a spanning tree of $\widetilde{G}$ which does not contain an edge with infinite length.

Elementary transition An elementary transformation corresponds to the modification of only one bit in the representation of a configuration. More precisely, let X_i be a configuration of the artificial edges of the graph $\widetilde{G}$. The binary representation of this configuration is $x_1^{(i)} x_2^{(i)} ... x_B^{(i)}$ where $B = |V \backslash R|$ and $x_k^{(i)} \in \{0, 1\}$ for $k = 1..., B$. We define the transformation m_l $(l = 1, ..., B)$ by

$$m_l(X_i) = X_{i+1} \quad \text{with} \quad x_j^{(i+1)} = \begin{cases} x_j^{(i)} & \text{si} \quad j \neq l \\ \\ x_j^{(i)} + 1 & \text{si} \quad j = l \end{cases}$$

Configurations in the neighbourhood of the current configuration X_i are those which are obtained by transformations m_l.

Let a_k indicate the artificial edge which corresponds to kth bit of a configuration and let $d(a_k)$ be its length.

An approximate solution according to the taboo search algorithm is presented in the following way:

```
// Initialisations
    For all edge c of G do
        D(c):= d(c); // save the original length
    Add artficial part to obtain the graph G~;
    X_0 = x_1^(0)...x_B^(0) with x_k = 0 (k = 1,...,B) // Initial configuration
    For k = 1,...,B do d(a_k):= ∞ EndDo
    T:= MST(G~) // minimal spanning tree of G~
    TabooList:= ∅
    i = 1
// Iterations
    While stop condition is false do
        For l = 1,...,B do
            If m_l ∉ TabooList then X_i:= m_l(X_{i-1}) EndIf
            If x_l^(i) = 1 then
                d(a_l) = 0
                For all adjacent edge c of a_l do
                    d(c):= ∞
              else
                d(a_l):= ∞
                For all adjacent edge c of a_l do
                    d(c):= D(c)
            EndIf
            TP_l:= MST(G~)
        EndDo
        TP:= min_l(TP_l) // seek the best successor
        X_i:= m_l(X_{i-1})
        If d(TP) < d(T) then
            T:= TP
          else
            Add m_l^{-1} to TabooList
        EndIf
        i:= i + 1
    EndDo
```

6 Numerical results

To analyze the quality of the suboptimal solutions can be achieved by using the new multicast routing method, it is necessary to make some notation about the network topologies and the method of performance analysis itself. But, before starting to describe the simulation process let us have a look at the routing background of the QOSPF [6, 11].

6.1 Technological background

The routers in each time instant (this time instant is set up by the network operator) advertise their link values (the available link bandwidth) over the network, in the following fashion (Figure 2):

- The parameter axis (the set of possible values of the available bandwidth) is covered with a grid $D = w_i, i = 1, ..., N$.
- At each time instant the value w_i is advertised from the link (u, v), if
  1. w_i has been exceeded by the available bandwidth on the link (u, v), or
  2. the actual value of the available bandwidth has been fallen below w_{i+1}.

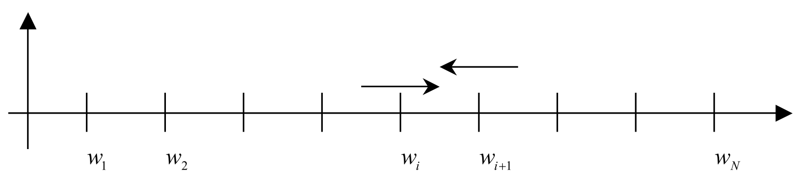

Figure 2: *The method for advertisement of the available link bandwidth in QOSPF routing environment*

So, the "public knowledge" about the value of the available link bandwidth ($\delta_{(u,v)}$) between node u and v is only the fact, that it is located somewhere in $[w_i, w_{i+1}]$. In current routing protocols the routers calculate routes (e.g. unicast paths, multicast trees) using the last advertised w_i values. More sophisticated utilisation of the network resources can be obtained by taking into account that the link parameter can take its value from the interval $[w_i, w_{i+1}]$ randomly. In our model this objective can be achieved by modelling the available bandwidth over the interval $[w_i, w_{i+1}]$ on link (u, v) by its probability distribution function $F^i_{(u,v)}(x)$ (Figure 3, assuming that w_i was the last advertised from link (u, v)).

6.2 Performance measures

It is obvious that the quality of the solution is determined by two factors:

1. the effect of reducing the original problem to the Steiner problem;

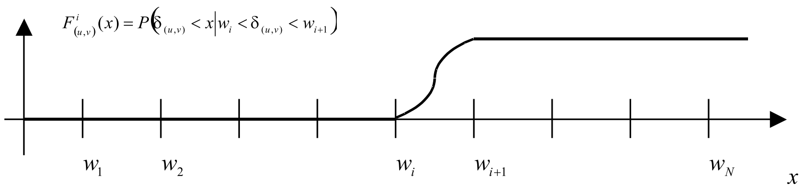

Figure 3: *Introducing probabilistic link measures*

2. the performance of the method (taboo search) used to find suboptimal solution for the Steiner tree problem;

The first type of errors is originated from the fact that the link independence hypothesis should be assumed to make the original problem (2) analytically tractable. In the following analysis we do not examine the effect of this assumption, we concentrate only on the second type of error. To analyze the performance of QoS routing based on bottleneck metric in incomplete environment the following spanning trees are introduced.

- T_{QOSPF} - can be calculated based on the last advertised $w_i^{(u,v)}$ link values T_{QOSPF} : $min_{(u,v)\in T} w_i^{(u,v)} > W$;
- $T_{opt-unknown}$ - can be calculated based on the randomly generated $\hat{\delta}_i^{(u,v)}$ link values $T_{opt-unknown}$: $min_{(u,v)\in T}\hat{\delta}_i^{(u,v)} > W$. This is the link values which are known only by the simulation process and unknown in real networking environment ($\hat{\delta}_i^{(u,v)}$ are generated in $[w_i, w_{i+1}]$ randomly according to $F^i_{(u,v)}(x)$);
- $T_{logprob}$ - can be calculated based on the $\kappa_{(u,v)} = -logP(\delta_{(u,v)} > W)$ link measures when approximating the optimal solution of (3).

When calculating the T_{QOSPF} or $T_{opt-unknown}$ spanning tree and at least one exists then the widest one is selected. To evaluate the performance of the taboo search method, the spanning tree $T_{logprob}$ is calculated by using the taboo search based method ($T_{logprob,taboo}$), the Kou algorithm [8] ($T_{logprob,Kou}$) and the Takahashi-Matsuyama algorithm [12] ($T_{logprob,T-M}$). The performance measure of a spanning tree T is the minimum amount of available capacity $min_{(u,v)\in T}\hat{\delta}_i^{(u,v)}$.

6.3 Network topologies and simulation parameters

To perform the simulation over real network dimensions the following network topology was used. Figure 4 depicts the nation wide backbone topology of the ANSNET and the modified network topology used by us to obtain simulation results.

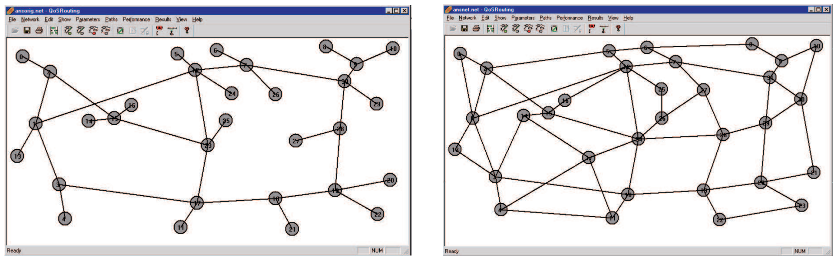

Figure 4: *The network topology of ANSNET and the modified topology*

Its connectivity is improved for ensuring the existence of multiple paths between each node pair [2, 3]. To get a realistic view about the operation of the multicast routing methods the bandwidth type QoS requirement and the network parameters were generated according to the distributions listed in the following table.

Parameter	Name	Value
W	QoS requirement	u. d. in $[0.1Mbit/s, 13Mbit/s]$
$\delta_{(u,v)}$	Random available bandwidth	u. d. in $[0Mbit/s, 15Mbit/s]$
$F^i_{(u,v)}(x)$	Conditional distribution	discrete u. d. in $[w_i^{(u,v)}, w_{i+1}^{(u,v)}]$
$w_{i+1}^{(u,v)} - w_i^{(u,v)}$	Grid granularity (equidistant)	3Mbit/s

6.4 Simulation process

The network topology during the simulation was not changed, (Figure 4), only its link values were generated randomly. Based on these parameters the simulation process can be described as:

1. Set $j = 1$.
2. Generate initial link values in the network depicted in Figure 4 according to the distribution of $\delta_{(u,v)}, \forall(u,v) \in E$.
3. Determine the interval $[w_i^{(u,v)}, w_{i+1}^{(u,v)}]$ on each link where the generated initial link value is located. This represents the situation that the link value $w_i^{(u,v)}$ was last advertised on link between nodes u and v (the only public knowledge in the network is that the exact link delay on (u,v) is somewhere in $[w_i^{(u,v)}, w_{i+1}^{(u,v)}]$).
4. Generate QoS requirement according to the distribution of W.
5. Generate the set of multicast nodes R_j.

6. Based on the generated QoS requirement, set of multicast nodes and the distribution of link parameters transform the probabilistic measures to deterministic ones (calculate $\kappa_{(u,v)}$) and find suboptimal solution for MLTS (obtain $T_{logprob,taboo}$, $T_{logprob,Kou}$, $T_{logprob,T-M}$).

7. Based on the $w_i^{(u,v)}$ values determine T_{QOSPF} (by calculating min-max spanning tree).

8. Generate the exact link values $\hat{\delta}_{(u,v)}$ in the interval $[w_i^{(u,v)}, w_{i+1}^{(u,v)})$ according to the distribution $F^i_{(u,v)}(x)$ and determine $T_{opt-unknown}$ (by calculating min-max spanning tree).

9. If $j < K$ then $j = j + 1$ and goto step 1.

The min-max spanning trees T_{QOSPF} and $T_{opt-unknown}$ were calculated by

1. Finding the path between each multicast node pair which has the maximum throughput (maximum minimal bandwidth). These path can be obtained by changing the operators of the Dijkstra algorithm (which is originally able to find shortest paths based on additive link metric) to the minimum and maximum operators;

2. Creating the metric closure of the original problem (fully connected graph where each node corresponds to a multicast node and each edge corresponds to a path found in step 1) and finding the maximal spanning tree by the Prim algorithm (the minimum operator is changed to the maximum operator). When the maximal spanning tree is found the paths included in the tree can be extracted and loops can be eliminated.

6.5 Simulation results

Based on the simulation process the five spanning trees (T_{QOSPF}, $T_{opt-unknown}$, $T_{logprob,taboo}$, $T_{logprob,Kou}$, $T_{logprob,T-M}$) were calculated at each randomly generated network. When these spanning trees are calculated the minimal capacity in the trees ($\min_{(u,v)\in T} \hat{\delta}_{(u,v)}$) are determined. The Figure 5 shows the minimal capacities of the spanning trees at each randomly generated network. The average values of the minimal capacities are determined over the randomly generated graph ensemble and depicted in Figure 6. While the figure shows the numerical difference between these values, the following table lists the percentages of ensuring the maximal minimum link capacity by the different type of multicast trees.

Multicast tree	Having the maximal minimum capacity
$T_{opt-unknown}$	100%
$T_{logprob,taboo}$	67.34%
$T_{logprob,Kou}$	63.26%
T_{QOSPF}	59.18%
$T_{logprob,T-M}$	57.14%

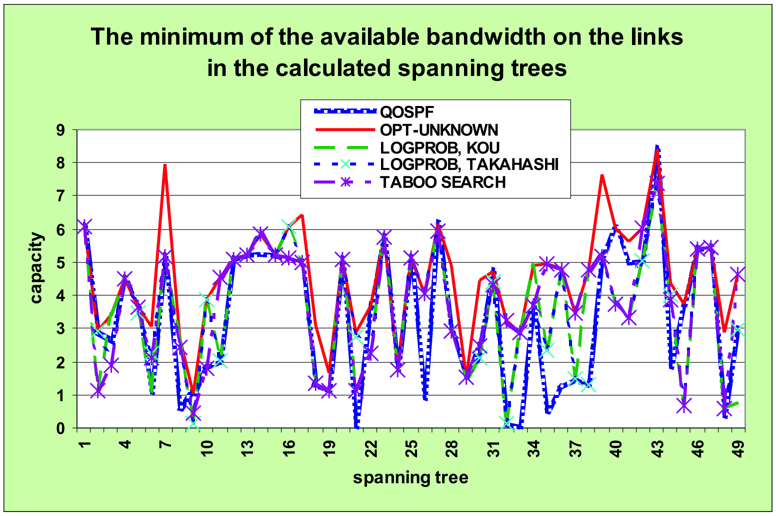

Figure 5: *The minimum of the available bandwidth on the links in the calculated spanning trees*

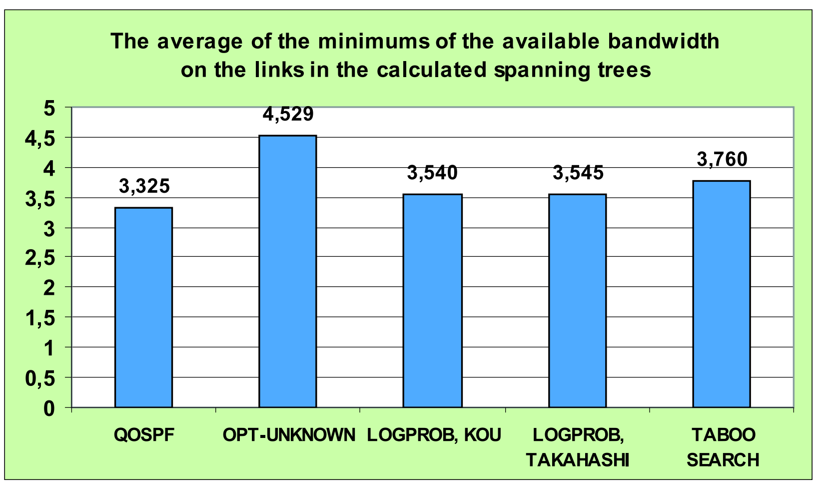

Figure 6: *The average of the minimums of the available bandwidth on the links in the calculated spanning trees*

The results confirms the fact that more suitable spanning trees could be calculated when the exact values of the available link capacities $\hat{\delta}_{(u,v)}$ are known (the average of the minimal link capacities has the highest value at $T_{opt-unknown}$). It is obvious that the optimality of this tree is related to the instant when calculations are performed. If the network state (the link load) changes, then the tree becommes probably less interesting. Originated from the fact that the values of the available link bandwidth are advertised at discrete time instants (the exact values are not known) and the spanning trees are calculated over these advertised link measures, the quality of the found trees can significantly be decreased (the the average of the minimal link capacities at T_{QOSPF} has lower value than at $T_{opt-unknown}$). Moreover, the results shows that the quality of these trees can be improved by introducing probabilistic link metrics and finding suboptimal solutions for the Steiner tree problem (the the average of the minimal link capacities at $T_{logprob,taboo}$, $T_{logprob,Kou}$ and $T_{logprob,T-M}$ has higher value than at T_{QOSPF}). Finally, the results shows that the quality of the suboptimal solutions can be found by the Kou or Takahashi-Matsuyama may be exceeded by the taboo search method.

7 Conclusions

A novel multicast routing algorithm was introduced in the paper based on the taboo search, which is able to meet bandwidth type QoS requirements even in the case of incomplete information. The routing based on bottleneck type of link metrics in incomplete environment was transformed into the Steiner Tree Problem and the algorithm yielded fast and a high performance routing. The simulation result showed that more appropriate multicast trees can be found by taking into account the probabilistic nature of the routing problem (originated from technical reasons, e.g. the value of the link descriptors can be advertised only at discrete time instants).

References

[1] A. Balakrishnan, N. R. Patel, *"Problem Reduction Methods and a Tree Generation Algorithm for the Steiner Network Problem"*, Networks, vol. 17. pp. 65-85, 1987.

[2] S. Chen, K. Nahrstedt: *"On Finding Multi-Constrained Paths"*, Proceedings of 7th IEEE International Conference on Communications, Atalanta, GA, pp. 874-879, June 1998.

[3] S. Chen, K. Nahrstedt: *"An overview of quality of service routing for next generation high-speed networks: Problems and solutions"*, IEEE Network Magazine, Special Issue on Transmission and Distribution of Digital Video, 12(6): 64-79, November-December 1998.

[4] H.N. Gabow, *"Two Algorithms for Generating Weighted Trees in Order"*, SIAM J. Comput, vol. 6. No 1. pp. 139-150, 1977.

[5] R. Guerin, A. Orda: *"QoS-based Routing in Networks with Inaccurate Information: Theory and Algorithms"*, IEEE INFOCOM'97, , Japan, April 1997.

[6] R. Guerin, A. Orda, D. Williams: *"QoS Routing Mechanisms and OSPF Extensions."*, Proceedings of 2nd Global Internet Miniconference, Phoenix, AZ, November 1997.

[7] F. K. Hwang, D. S. Richards, *"Steiner Tree Problems"*, Computer Science Report No. TR-87-21, University of Virginia, 1987.

[8] L. Kou, G. Markowsky, L. Berman: *"A Fast Algorithm for Steiner Tree"*, Acta Informatica 15, pp. 141-145, 1981.

[9] D. H. Lorenz, A. Orda: *"QoS Routing in Networks with Uncertain Parameters"*, IEEE INFOCOM'98, , Japan, March 1998.

[10] M. Molnár, R. Marie, *"Forêts déductibles maximales"*, RR INRIA no 3974, mars 1999, URL= ftp://ftp.inria.fr/INRIA/publication/RR/RR-3974.ps.gz

[11] G. Apostolopoulos, D. Williams, S. Kamat, R. Guerin, A. Orda, T. Przygienda: *"QoS Routing Mechanisms and OSPF Extensions."*, RFC 2676, www.ietf.org, August 1999.

[12] H. Takahashi, A. Matsuyama: *"An Approximate Solution for the Steiner Tree Problem in Graphs"*, Mathematica Japonica, 1980.

[13] P. Winter, *"Steiner problem in networks: A survey"*, Networks, vol. 17. pp. 129-167, 1987.

[14] J. Xu, Steve Y. Chiu, F. Glover *Tabu search for dynamic routing communications network design*, Telecommunicatio Systems, vol. 8. pp 55-77, 1997.

[15] J. Xu, Steve Y. Chiu, F. Glover *Optimizing a Ring-Based Priivate Line Telecommunication Network using Tabu Search*, Management Science, vol. 45, No 3, March 1999.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399